分布式系统概论 Lab2

指导老师：冯红伟

November 24, 2023

Contents

设计 `MapReduce` 程序，实现倒排文档索引。

1 倒排文档索引

倒排文档索引 TF-IDF（term frequency–inverse document frequency）是一种用于信息检索与数据挖掘的常用加权技术。TF 是词频 (Term Frequency)，IDF 是逆文本频率指数 (Inverse Document Frequency)。[1]倒排文档索引常用于关键词搜索，一个词语在一篇文章中出现次数越多, 同时在所有文档中出现次数越少, 则越能够代表该文章。

TF-IDF 的计算可分解为三步。

第一步，计算词频 TF：

$$TF_w = \frac{N_w}{N}$$

其中 N_w 是在某一文档中词条 w 出现的次数，N 是该文档总词条数。

第二步，计算逆向文件频率 IDF：

$$IDF_w = log(\frac{Y}{Y_w + 1})$$

其中 Y 是语料库的文档总数，Y_w 是包含词条 w 的文档数，分母加 1 是为了避免 w 未出现在任何文档中从而导致分母为 0 的情况。

第三步，计算倒排文档索引 TF-IDF：

$$TF - IDF_w = TF_w * IDF_w$$

2 实现

基于 TF-IDF 模型，你需要实现以下功能：

使用作业提供的文本数据 document.dat，实现根据查询关键词，返回相关文档名（序列）的功能。

不需要设计程序界面，要求实现倒排文档生成函数和文档查询函数。提交代码文件和查询返回结果的示例截屏。

2.1 mrjob

`mrjob` 是由 `Yelp` 创建的 Python MapReduce 库，允许用户在未安装 hadoop 或部署集群的情况下执行 MapReduce 程序。

[1]参考学习 https://www.jianshu.com/p/091383e86825

mrjob 可以使用 pip 进行安装：

```
# 安装 mrjob
>> pip install mrjob
```

通过继承父类 MRJob，可以实现定制的 MapReduce 实例，以下是一个简单的英文文本词数统计示例：

```python
from mrjob.job import MRJob

class WordCount(MRJob):
    # 定义mapper
    def mapper(self, _, line):
        yield "words", len(line.split())

    # 定义reducer
    def reducer(self, key, values):
        yield key, sum(values)

if __name__ == '__main__':
    WordCount.run()
```

假设上述代码文件为 wordcount.py,需要处理的数据文件为 data.txt，则可以使用以下指令将词数统计结果输入到文件 result.txt 中：

```
# 执行 MR 程序
>> python wordcount.py data.txt >> result.txt
```

2.2 文件读取和分词

待处理的文本数据 document.dat 使用 gb18030 编码格式，可以使用以下函数打开：

```
# 读取文件
open('document.dat','r', encoding='gb18030')
```

python 中，可以使用 jieba 包对中文内容分词：

```
# 分词
import jieba
seg_list = jieba.cut_for_search(" 倒排文档索引是一种统计方法，用以评估字词对于语料库中其中一份文件的重要程度")
```

3 上传

3.1 截止日期

6/12/2023 23:59(GMT+8)

3.2 提交内容

1. 实验报告（40%），介绍你的设计思路和实现，展示运行结果截图。包括
   - 数据预处理：由`.dat`文件生成可供`MapReduce`程序处理的`.txt`文件
   - TF-IDF 文档生成：生成有 TF-IDF 文档。每条记录对应一个关键词，采用 Keyword [[Document1, TF-IDF1], [Document2, TF-IDF2], ...] 的格式
   - 文档查询：输入查询关键词，返回相关文档名（序列）
2. 实现代码（60%），完成数据预处理、TF-IDF 文档生成和文档查询函数。